

# Cluster Forests

Donghui Yan<sup>†\*</sup>, Aiyou Chen<sup>‡</sup>, Michael I. Jordan<sup>§\*</sup>

<sup>†</sup>Department of Statistics

<sup>§</sup>Department of Statistics and Department of EECS

<sup>\*</sup>University of California, Berkeley, CA 94720

<sup>‡</sup>Bell Labs, Alcatel-Lucent, Murray Hill, NJ 07974

## Abstract

Inspired by Random Forests (RF) in the context of classification, we propose a new clustering ensemble method—Cluster Forests (CF). Geometrically, CF randomly probes a high-dimensional data cloud to obtain “good local clusterings” and then aggregates via spectral clustering to obtain cluster assignments for the whole dataset. The search for good local clusterings is guided by a cluster quality measure  $\kappa$ . CF progressively improves each local clustering in a fashion that resembles the tree growth in RF. Empirical studies on several real-world datasets under two different performance metrics show that CF compares favorably to its competitors. Theoretical analysis shows that the  $\kappa$  criterion is shown to grow each local clustering in a desirable way—it is “noise-resistant.” A closed-form expression is obtained for the mis-clustering rate of spectral clustering under a perturbation model, which yields new insights into some aspects of spectral clustering.

## 1 Motivation

The general goal of clustering is to partition a set of data such that data points within the same cluster are “similar” while those from different clusters are “dissimilar.” An emerging trend is that new applications tend to generate data in very high dimensions for which traditional methodologies of cluster analysis do not work well. Remedies include dimension reduction and feature transformation, but it is a challenge to develop effective instantiations of these remedies in the high-dimensional clustering setting. In particular, in datasets whose dimension is beyond 20, it is infeasible to perform full subset selection. Also, there may not be a single set of attributes on which the whole set of data can be reasonably separated. Instead, there may be local patterns in which different choices of attributes or different projections reveal the clustering.

---

<sup>1</sup>The authors can be contacted via email at: dhyan@berkeley.edu, aiyouchen@google.com, jordan@eecs.berkeley.edu. Aiyou Chen is currently with Google, Mountain View, CA 94043.

Our approach to meeting these challenges is to randomly probe the data/feature space to detect many locally “good” clusterings and then aggregate by spectral clustering. The intuition is that, in high-dimensional spaces, there may be projections or subsets of the data are well separated and these projections or subsets may carry information about the cluster membership of the data involved. If we can effectively combine such information from many different views (here a view has two components, the directions or projections we are looking at and the part of data that are involved), then we can hope to recover the cluster assignments for the whole dataset. However, the number of projections or subsets that are potentially useful tend to be huge, and it is not feasible to conduct a grand tour of the whole data space by exhaustive search. This motivates us to randomly probe the data space and then aggregate.

The idea of random projection has been explored in various problem domains such as clustering [12, 9], manifold learning [19], and compressive sensing [10]. However, the most direct motivation for our work is the Random Forests (RF) methodology for classification [7]. In RF, a *bootstrap step* selects a subset of data while the *tree growth step* progressively improves a tree from the root downwards—each tree starts from a random collection of variables at the root and then becomes stronger and stronger as more nodes are split. Similarly, we expect that it will be useful in the context of high-dimensional clustering to go beyond simple random probings of the data space and to perform a controlled probing in hope that most of the probings are “strong.” This is achieved by progressively refining our “probings” so that eventually each of them can produce relatively high-quality clusters although they may start weak. In addition to the motivation from RF, we note that similar ideas have been explored in the projection pursuit literature for regression analysis and density estimation (see [22] and references therein).

RF is a supervised learning methodology and as such there is a clear goal to achieve, i.e., good classification or regression performance. In clustering, the goal is less apparent. But significant progress has been made in recent years in treating clustering as an optimization problem under an explicitly defined cost criterion; most notably in the spectral clustering methodology [39, 42]. Using such criteria makes it possible to develop an analog of RF in the clustering domain.

Our contributions can be summarized as follows. We propose a new cluster ensemble method that incorporates model selection and regularization. Empirically CF compares favorably to some popular cluster ensemble methods. We also provide some theoretical support for our work: (1) Under a simplified model, CF is shown to grow the clustering instances in a “noise-resistant” manner; (2) we obtain a closed-form formula for the mis-clustering rate of spectral clustering under a perturbation model that yields new insights into aspects of spectral clustering that are relevant to CF.

The remainder of the paper is organized as follows. In Section 2, we present a detailed description of CF. Related work is discussed in Section 3. This is followed by an analysis of the  $\kappa$  criterion and the mis-clustering rate of spectral clustering under a perturbation model in Section 4. We evaluate our method

in Section 5 by simulations on Gaussian mixtures and comparison to several popular cluster ensemble methods on real-world data. Finally we conclude in Section 6.

## 2 The Method

CF is an instance of the general class of *cluster ensemble methods* [36], and as such it is comprised of two phases: one which creates many cluster instances and one which aggregates these instances into an overall clustering. We begin by discussing the cluster creation phase.

### 2.1 Growth of clustering vectors

CF works by aggregating many instances of clustering problems, with each instance based on a different subset of features (with varying weights). We define the *feature space*  $\mathcal{F} = \{1, 2, \dots, p\}$  as the set of indices of coordinates in  $\mathbb{R}^p$ . We assume that we are given  $n$  i.i.d. observations  $X_1, \dots, X_n \in \mathbb{R}^p$ . A *clustering vector* is defined to be a subset of the feature space.

The growth of a clustering vector is governed by the following cluster quality measure:

$$\kappa(\tilde{\mathbf{f}}) = \frac{SS_W(\tilde{\mathbf{f}})}{SS_B(\tilde{\mathbf{f}})} \quad (1)$$

where  $SS_W$  and  $SS_B$  are the within-cluster and between-cluster sum of squared distances (see Section 7.2), computed on the set of features currently in use (denoted by  $\tilde{\mathbf{f}}$ ), respectively.

Using this quality measure, we iteratively expand the clustering vector. Specifically, letting  $\tau$  denote the number of consecutive unsuccessful attempts in expanding the clustering vector  $\tilde{\mathbf{f}}$ , and letting  $\tau_m$  be the maximal allowed value of  $\tau$ , the growth of a clustering vector is described in Algorithm 1.

---

**Algorithm 1** The growth of a clustering vector  $\tilde{\mathbf{f}}$

---

- 1: Initialize  $\tilde{\mathbf{f}}$  to be NULL and set  $\tau = 0$ .
  - 2: Apply feature competition and update  $\tilde{\mathbf{f}} \leftarrow (f_1^{(0)}, \dots, f_b^{(0)})$ .
  - 3: **repeat**
  - 4:   Sample  $b$  features, denoted as  $f_1, \dots, f_b$ , from the feature space  $\mathcal{F}$ .
  - 5:   Apply  $K$ -means (the *base clustering algorithm*) to the data induced by the feature vector  $(\tilde{\mathbf{f}}, f_1, \dots, f_b)$ .
  - 6:   **if**  $\kappa(\tilde{\mathbf{f}}, f_1, \dots, f_b) < \kappa(\tilde{\mathbf{f}})$  **then**
  - 7:     expand  $\tilde{\mathbf{f}}$  by  $\tilde{\mathbf{f}} \leftarrow (\tilde{\mathbf{f}}, f_1, \dots, f_b)$  and set  $\tau \leftarrow 0$
  - 8:   **else**
  - 9:     discard  $\{f_1, \dots, f_b\}$  and set  $\tau \leftarrow \tau + 1$
  - 10:   **end if**
  - 11: **until**  $\tau \geq \tau_m$
-

---

**Algorithm 2** Feature competition

---

- 1: **for**  $i = 1$  **to**  $q$  **do**
  - 2:   Sample  $b$  features,  $f_1^{(i)}, \dots, f_b^{(i)}$ , from the feature space  $\mathcal{F}$ ;
  - 3:   Apply  $K$ -means to the data projected on  $(f_1^{(i)}, \dots, f_b^{(i)})$  to get  $\kappa(f_1^{(i)}, \dots, f_b^{(i)})$ ;
  - 4: **end for**
  - 5: Set  $(f_1^{(0)}, \dots, f_b^{(0)}) \leftarrow \arg \min_{i=1}^q \kappa(f_1^{(i)}, \dots, f_b^{(i)})$ .
- 

Step 2 in Algorithm 1 is called *feature competition* (setting  $q = 1$  reduces to the usual mode). It aims to provide a good initialization for the growth of a clustering vector. The feature competition procedure is detailed in Algorithm 2.

Feature competition is motivated by Theorem 1 in Section 4.1—it helps prevent noisy or “weak” features from entering the clustering vector at the initialization, and, by Theorem 1, the resulting clustering vector will be formed by “strong” features which can lead to a “good” clustering instance. This will be especially helpful when the number of noisy or very weak features is large. Note that feature competition can also be applied in other steps in growing the clustering vector. A heuristic for choosing  $q$  is based on the “feature profile plot,” a detailed discussion of which is provided in Section 5.2.

## 2.2 The CF algorithm

The CF algorithm is detailed in Algorithm 3. The key steps are: (a) grow  $T$  clustering vectors and obtain the corresponding clusterings; (b) average the clustering matrices to yield an aggregate matrix  $P$ ; (c) regularize  $P$ ; and (d) perform spectral clustering to the regularized matrix. The regularization step is done by *thresholding*  $P$  at level  $\beta_2$ ; that is, setting  $P_{ij}$  to be 0 if it is less than a constant  $\beta_2 \in (0, 1)$ , followed by a further nonlinear transformation  $P \leftarrow \exp(\beta_1 P)$  which we call *scaling*.

---

**Algorithm 3** The CF algorithm

---

- 1: **for**  $l = 1$  **to**  $T$  **do**
- 2:   Grow a clustering vector,  $\tilde{\mathbf{f}}^{(l)}$ , according to Algorithm 1;
- 3:   Apply the base clustering algorithm to the data induced by clustering vector  $\tilde{\mathbf{f}}^{(l)}$  to get a partition of the data;
- 4:   Construct  $n \times n$  co-cluster indicator matrix (or affinity matrix)  $P^{(l)}$

$$P_{ij}^{(l)} = \begin{cases} 1, & \text{if } X_i \text{ and } X_j \text{ are in the same cluster} \\ 0, & \text{otherwise} \end{cases}$$

- 5: **end for**
  - 6: Average the indicator matrices to get  $P \leftarrow \frac{1}{T} \sum_{l=1}^T P^{(l)}$ ;
  - 7: Regularize the matrix  $P$ ;
  - 8: Apply spectral clustering to  $P$  to get the final clustering.
-

We provide some justification for our choice of spectral clustering in Section 4.2. As the entries of matrix  $P$  can be viewed as encoding the pairwise similarities between data points, any clustering algorithm based on pairwise similarity can be used as the aggregation engine.

### 3 Related Work

In this section, we compare and contrast CF to other work on cluster ensembles. It is beyond our scope to attempt a comprehensive review of the enormous body of work on clustering, please refer to [23, 18] for overview and references. We will also omit a discussion on classifier ensembles, see [18] for references. Our focus will be on cluster ensembles. We discuss the two phases of cluster ensembles, namely, the generation of multiple clustering instances and their aggregation, separately.

For the generation of clustering instances, there are two main approaches—data re-sampling and random projection. [11] and [28] produce clustering instances on bootstrap samples from the original data. Random projection is used by [12] where each clustering instance is generated by randomly projecting the data to a lower-dimensional subspace. These methods are myopic in that they do not attempt to use the quality of the resulting clusterings to choose samples or projections. Moreover, in the case of random projections, the choice of the dimension of the subspace is myopic. In contrast, CF proceeds by selecting features that progressively improve the quality (measured by  $\kappa$ ) of individual clustering instances in a fashion resembling that of RF. As individual clustering instances are refined, better final clustering performance can be expected. We view this non-myopic approach to generating clustering instances as essential when the data lie in a high-dimensional ambient space. Another possible approach is to generate clustering instances via random restarts of a base clustering algorithm such as  $K$ -means [14].

The main approaches to aggregation of clustering instances are the *co-association method* [36, 15] and the *hyper-graph method* [36]. The co-association method counts the number of times two points fall in the same cluster in the ensemble. The hyper-graph method solves a  $k$ -way minimal cut hyper-graph partitioning problem where a vertex corresponds to a data point and a link is added between two vertices each time the two points meet in the same cluster. Another approach is due to [37], who propose to combine clustering instances with mixture modeling where the final clustering is identified as a maximum likelihood solution. CF is based on co-association, specifically using spectral clustering for aggregation. Additionally, CF incorporates regularization such that the pairwise similarity entries that are close to zero are thresholded to zero. This yields improved clusterings as demonstrated by our empirical studies.

A different but closely related problem is clustering aggregation [17] which requires finding a clustering that “agrees” as much as possible with a given set of input clustering instances. Here these clustering instances are assumed

to be known and the problem can be viewed as the second stage of clustering ensemble. Also related is ensemble selection [8, 13, 5] which is applicable to CF but this is not the focus of the present work. Finally there is unsupervised learning with random forests [7, 35] where RF is used for deriving a suitable distance metric (by synthesizing a copy of the data via randomization and using it as the “contrast” pattern) so they are fundamentally different from ours.

## 4 Theoretical Analysis

In this section, we provide a theoretical analysis of some aspects of CF. In particular we develop theory for the  $\kappa$  criterion, presenting conditions under which CF is “noise-resistant.” By “noise-resistant” we mean that the algorithm can prevent a pure noise feature from entering the clustering vector. We also present a perturbation analysis of spectral clustering, deriving a closed-form expression for the mis-clustering rate.

### 4.1 CF is noise-resistant

We analyze the case in which the clusters are generated by a Gaussian mixture:

$$\Delta \mathcal{N}(\boldsymbol{\mu}, \Sigma) + (1 - \Delta) \mathcal{N}(-\boldsymbol{\mu}, \Sigma), \quad (2)$$

where  $\Delta \in \{0, 1\}$  with  $\mathbb{P}(\Delta = 1) = \pi$  specifies the cluster membership of an observation, and  $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$  stands for a Gaussian random variable with mean  $\boldsymbol{\mu} = (\mu[1], \dots, \mu[p]) \in \mathbb{R}^p$  and covariance matrix  $\Sigma$ . We specifically consider  $\pi = \frac{1}{2}$  and  $\Sigma = \mathbf{I}_{p \times p}$ ; this is a simple case which yields some insight into the feature selection ability of CF. We start with a few definitions.

**Definition.** Let  $h : \mathbb{R}^p \mapsto \{0, 1\}$  be a *decision rule*. Let  $\Delta$  be the cluster membership for observation  $X$ . A loss function associated with  $h$  is defined as

$$l(h(X), \Delta) = \begin{cases} 0, & \text{if } h(X) = \Delta \\ 1, & \text{otherwise.} \end{cases} \quad (3)$$

The optimal clustering rule under (3) is defined as

$$h^* = \arg \min_{h \in \mathcal{G}} \mathbb{E} l(h(X), \Delta), \quad (4)$$

where  $\mathcal{G} \triangleq \{h : \mathbb{R}^p \mapsto \{0, 1\}\}$  and the expectation is taken with respect to the random vector  $(X, \Delta)$ .

**Definition [32].** For a probability measure  $Q$  on  $\mathbb{R}^d$  and a finite set  $A \subseteq \mathbb{R}^d$ , define the *within cluster sum of distances* by

$$\Phi(A, Q) = \int \min_{a \in A} \phi(\|x - a\|) Q(dx) \quad (5)$$

where  $\phi(\|x - a\|)$  defines the distance between points  $x$  and  $a \in A$ .  $K$ -means clustering seeks to minimize  $\Phi(A, Q)$  over a set  $A$  with at most  $K$  elements. We

focus on the case  $\phi(x) = x^2$ ,  $K = 2$  and refer to  $\{\mu_0^*, \mu_1^*\} = \underset{A}{\operatorname{argmin}} \Phi(A, Q)$  as the *population cluster centers*.

**Definition.** The  $i^{\text{th}}$  feature is called a *noise* feature if  $\mu[i] = 0$  where  $\mu[i]$  denotes the  $i^{\text{th}}$  coordinate of  $\boldsymbol{\mu}$ . A feature is “*strong*” (“*weak*”) if  $|\mu[i]|$  is “large” (“small”).

**Theorem 1.** Assume the cluster is generated by Gaussian mixture (2) with  $\Sigma = \mathbf{I}$  and  $\pi = \frac{1}{2}$ . Assume one feature is considered at each step and duplicate features are excluded. Let  $\mathcal{I} \neq \emptyset$  be the set of features currently in the clustering vector and let  $f_n$  be a noise feature such that  $f_n \notin \mathcal{I}$ . If  $\sum_{i \in \mathcal{I}} (\mu_0^*[i] - \mu_1^*[i])^2 > 0$ , then  $\kappa(\mathcal{I}) < \kappa(\{\mathcal{I}, f_n\})$ .

**Remark.** The interpretation of Theorem 1 is that noise features are generally not included in cluster vectors under the CF procedure; thus, CF with the  $\kappa$  criterion is noise-resistant.

The proof of Theorem 1 is in the appendix. It proceeds by explicitly calculating  $SS_B$  and  $SS_W$  (see Section 7.2) and thus an expression for  $\kappa = SS_W/SS_B$ . The calculation is facilitated by the equivalence, under  $\pi = \frac{1}{2}$  and  $\Sigma = \mathbf{I}$ , of  $K$ -means clustering and the optimal clustering rule  $h^*$  under loss function (3).

## 4.2 Quantifying the mis-clustering rate

Recall that spectral clustering works on a weighted similarity graph  $\mathcal{G}(\mathbf{V}, P)$  where  $\mathbf{V}$  is formed by a set of data points  $X_i, i = 1, \dots, n$  and  $P$  encodes their pairwise similarities. Spectral clustering algorithms compute the eigendecomposition of the Laplacian matrix  $\mathcal{L}(P) = D^{-1/2}(I - P)D^{-1/2}$  where  $D$  is a diagonal matrix with diagonals being degrees of  $\mathcal{G}$ . Different notions of similarity and ways of using the spectral decomposition lead to different spectral cluster algorithms [34, 26, 30, 24, 39, 42]. In particular, Ncut [34] forms a bipartition of the data according to the sign of the components of the second eigenvector (i.e., corresponding to the second smallest eigenvalue) of  $\mathcal{L}(P)$ . On each of the two partitions, Ncut is then applied recursively until a stopping criterion is met.

There has been relatively little theoretical work on spectral clustering; exceptions include [4, 30, 24, 38, 29, 1, 40]. Here we analyze the mis-clustering rate for symmetrically normalized spectral clustering. For simplicity we consider the case of two clusters under a perturbation model.

Assume that the similarity (affinity) matrix can be written as

$$P = \overline{P} + \varepsilon, \quad (6)$$

where

$$\overline{P}_{ij} = \begin{cases} 1 - \nu, & \text{if } i, j \leq n_1 \text{ or } i, j > n_1 \\ \nu, & \text{otherwise.} \end{cases}$$

and  $\varepsilon = (\varepsilon_{ij})_1^n$  is a symmetric random matrix such that  $\mathbb{E}\varepsilon_{ij} = 0$ . Here  $n_1$  and  $n_2$  are the size of the two clusters. Let  $n_2 = \gamma n_1$  and  $n = n_1 + n_2$ . Without loss of generality, assume  $\gamma \leq 1$ . Similar models have been studied in earlier

work; see, for instance, [20, 31, 2, 6]. Our focus is different; we aim at the mis-clustering rate due to perturbation. Such a model is appropriate for modeling the similarity (affinity) matrix produced by CF. For example, Figure 1 shows the affinity matrix produced by CF on the Soybean dataset [5]; this matrix is nearly block-diagonal with each block corresponding to data points from the same cluster (there are totally 4 of them) and the off-diagonal elements are mostly "close" to 0. Thus a perturbation model as (6) is a "good" approximation to the similarity matrix produced by CF and allows us to gain insights into the nature of CF.

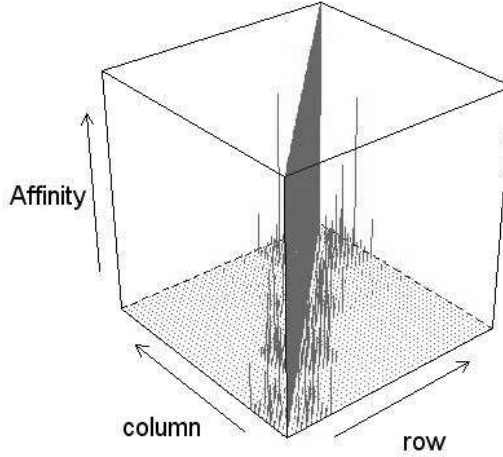


Figure 1: The affinity matrix produced by CF for the Soybean dataset with 4 clusters. The number of clustering vectors in the ensemble is 100.

Let  $\mathcal{M}$  be the mis-clustering rate, i.e., the proportion of data points assigned to a wrong cluster (i.e.,  $h(X) \neq \Delta$ ). Theorem 2 characterizes the expected value of  $\mathcal{M}$  under perturbation model (6).

**Theorem 2.** *Assume that  $\varepsilon_{ij}$ ,  $i \geq j$  are mutually independent  $\mathcal{N}(0, \sigma^2)$ . Let  $0 < \nu \ll \gamma \leq 1$ . Then*

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log(\mathbb{E}\mathcal{M}) = -\frac{\gamma^2}{2\sigma^2(1+\gamma)(1+\gamma^3)}. \quad (7)$$

The proof is in the appendix. The main step is to obtain an analytic expression for the second eigenvector of  $\mathcal{L}(P)$ . Our approach is based on matrix perturbation theory [25], and the key idea is as follows.



Let  $\mathfrak{P}(A)$  denote the eigenprojection of a linear operator  $A : \mathbb{R}^n \rightarrow \mathbb{R}^n$ . Then,  $\mathfrak{P}(A)$  can be expressed explicitly as the following contour integral

$$\mathfrak{P}(A) = \frac{1}{2\pi i} \oint_{\Gamma} (A - \zeta I)^{-1} d\zeta \quad (8)$$

where  $\Gamma$  is a simple Jordan curve enclosing the eigenvalues of interest (i.e., the first two eigenvalues of matrix  $\mathcal{L}(P)$ ) and excluding all others. The eigenvectors of interest can then be obtained by

$$\varphi_k = \mathfrak{P}\omega_i, \quad i = 1, 2 \quad (9)$$

where  $\omega_i, i = 1, 2$  are fixed linearly independent vectors in  $\mathbb{R}^n$ . An explicit expression for the second eigenvector can then be obtained under perturbation model (6), which we use to calculate the final mis-clustering rate.

**Remarks.** While formula (7) is obtained under some simplifying assumptions, it provides insights into the nature of spectral clustering.

- 1). The mis-clustering rate increases as  $\sigma$  increases.
- 2). By checking the derivative, the right hand side of (7) is a unimodal function of  $\gamma$ , minimized at  $\gamma = 1$  with a fixed  $\sigma$ . Thus the mis-clustering rate decreases as the cluster sizes become more balanced.
- 3). When  $\gamma$  is very small, i.e., the clusters are extremely unbalanced, spectral clustering is likely to fail.

These results are consistent with existing empirical findings. In particular, they underscore the important role played by the ratio of two cluster sizes,  $\gamma$ , on the mis-clustering rate. Additionally, our analysis (in the proof of Theorem 2) also implies that the best cutoff value (when assigning cluster membership based on the second eigenvector) is not exactly 0 but shifts slightly towards the center of those components of the second eigenvector that correspond to the smaller cluster. A closely related work is [21] which studies end-to-end perturbation and the final mis-clustering rate is *approximate* in nature. Theorem 2 is based on a perturbation model for the affinity matrix and provides, for the first time, a closed-form expression for the mis-clustering rate of spectral clustering under such a model.

## 5 Experiments

Two sets of experiments are performed, one on synthetic data, specifically designed to demonstrate the feature selection and “noise-resistance” capability of CF, and the other on several real-world datasets [3] where we compare the overall clustering performance of CF with several competitors under two different metrics. These experiments are presented in separate subsections.

### 5.1 Feature selection capability of CF

In this subsection, we describe three simulations that aim to study the feature selection capability and “noise-resistance” feature of CF. Assume the underlying data are generated i.i.d. by Gaussian mixture (2).

In the first simulation, a sample of 4000 observations is generated from (2) with  $\boldsymbol{\mu} = (0, \dots, 0, 1, 2, \dots, 100)^T$  and the diagonals of  $\Sigma$  are all 1 while the non-diagonals are i.i.d. uniform from  $[0, 0.5]$  subject to symmetry and positive definiteness of  $\Sigma$ . Denote this dataset as  $\mathbf{G}_1$ . At each step of cluster growing one feature is sampled from  $\mathcal{F}$  and tested to see if it is to be included in the clustering vector by the  $\kappa$  criterion. We run the clustering vector growth procedure until all features have been attempted with duplicate features excluded. 100 clustering vectors are generated. In Figure 2, all but one of the 100 clustering vectors include at least one feature from the top 3 features (ranked according to the  $|\mu[i]|$  value) and all clustering vectors contain at least one of the top 5 features.

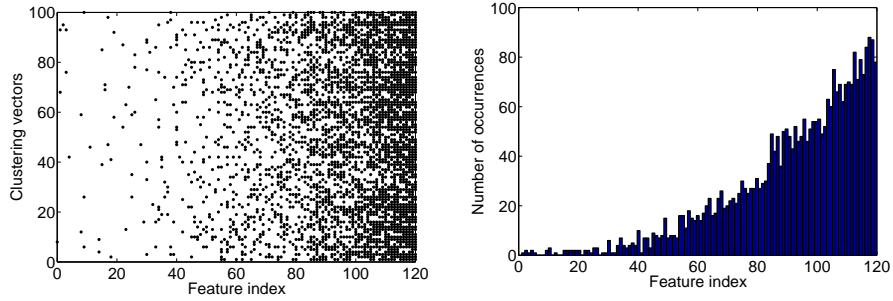


Figure 2: The occurrence of individual features in the 100 clustering vectors for  $\mathbf{G}_1$ . The left plot shows the features included (indicated by a solid circle) in each clustering vector. Each horizontal line corresponds to a clustering vector. The right plot shows the total number of occurrences of each feature.

We also performed a simulation with “noisy” data. In this simulation, data are generated from (2) with  $\Sigma = \mathbf{I}$ , the identity matrix, such that the first 100 coordinates of  $\boldsymbol{\mu}$  are 0 and the next 20 are generated i.i.d. uniformly from  $[0, 1]$ . We denote this dataset as  $\mathbf{G}_2$ . Finally, we also considered an extreme case where data are generated from (2) with  $\Sigma = \mathbf{I}$  such that the first 1000 features are noise features and the remaining 20 are useful features (with coordinates of  $\mu$  from  $\pm 1$  to  $\pm 20$ ); this is denoted as  $\mathbf{G}_3$ . The occurrences of individual features for  $\mathbf{G}_2$  and  $\mathbf{G}_3$  are shown in Figure 3. Note that the two plots in Figure 3 are produced by invoking feature competition with  $q = 20$  and  $q = 50$ , respectively. It is worthwhile to note that, for both  $\mathbf{G}_2$  and  $\mathbf{G}_3$ , despite the fact that a majority of features are pure noise (100 out of a total of 120 for  $\mathbf{G}_2$  or 1000 out of 1020 for  $\mathbf{G}_3$ , respectively), CF achieves clustering accuracies (computed against the true labels) that are very close to the Bayes rates (about 1).

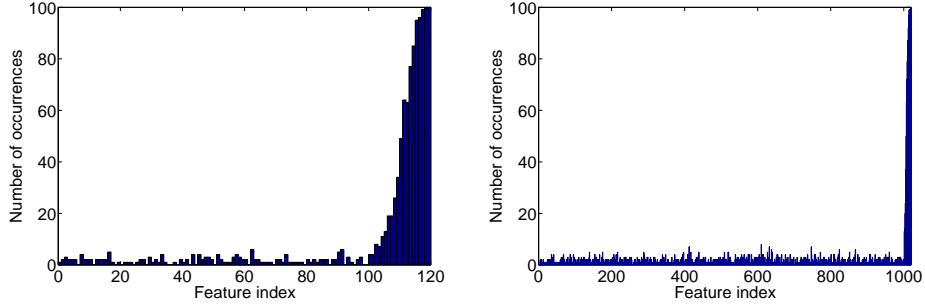


Figure 3: The occurrence of individual features in the 100 clustering vectors for  $G_2$  and  $G_3$ . The left plot is for  $G_2$  where the first 100 features are noise features. The right plot is for  $G_3$  where the first 1000 features are noise ones.

## 5.2 Experiments on UC Irvine datasets

We conducted experiments with six UC Irvine datasets [3], the Soybean, SPECT Heart, image segmentation (ImgSeg), Heart, Wine and Wisconsin breast cancer (WDBC) datasets. A summary is provided in Table 1. Note that true labels are available for all six datasets. We use the labels to evaluate the performance of the clustering methods, while recognizing that this evaluation is only partially satisfactory. The evaluation is based on two different performance metrics for clustering,  $\rho_r$  and  $\rho_c$ , to be defined in the following.

Dataset	Features	Classes	#Instances
Soybean	35	4	47
SPECT	22	2	267
ImgSeg	19	7	2100
Heart	13	2	270
Wine	13	3	178
WDBC	30	2	569

Table 1: A summary of datasets.

**Definition.** One measure of the quality of a cluster ensemble is given by

$$\rho_r = \frac{\text{Number of correctly clustered pairs}}{\text{Total number of pairs}} \times 100\%$$

where by “correctly clustered pair” we mean two instances have the same co-cluster membership (that is, they are in the same cluster) by both CF and the labels in the original dataset.

**Definition.** Another performance metric is the clustering accuracy. Let  $z = \{1, 2, \dots, J\}$  denote the set of class labels, and  $\theta(.)$  and  $f(.)$  the true label

and the label obtained by a clustering algorithm, respectively. The clustering accuracy is defined as

$$\rho_c(f) = \max_{\tau \in \Pi_{\mathbf{z}}} \left\{ \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{\tau(f(X_i)) = \theta(X_i)\} \right\}, \quad (10)$$

where  $\mathbb{I}$  is the indicator function and  $\Pi_{\mathbf{z}}$  is the set of all permutations on the label set  $\mathbf{z}$ . This measure is a natural extension of the classification accuracy (under 0-1 loss) and has been used by a number of work in clustering, e.g., [27, 41].

The idea of having two different performance metrics is to assess a clustering algorithm from different perspectives since one metric may particularly favor certain aspects while overlooking others. For example, in our experiment we observe that, for some datasets, some clustering algorithms (e.g., RP or EA) achieve a high value of  $\rho_r$  but a small  $\rho_c$  on the same clustering instance (note that  $\rho_c$  and  $\rho_r$  as reported in Table 2 and Table 3 may be calculated under different parameter settings).

Dataset	CF	RP	bC2	EA
Soybean	<b>92.36</b>	87.04	83.16	86.48
SPECT	<b>56.78</b>	49.89	50.61	51.04
ImgSeg	79.71	<b>85.88</b>	82.19	85.75
Heart	<b>56.90</b>	52.41	51.50	53.20
Wine	<b>79.70</b>	71.94	71.97	71.86
WDBC	<b>79.66</b>	74.89	74.87	75.04

Table 2:  $\rho_r$  for different datasets and methods (CF calculated when  $q = 1$ ).

Dataset	CF	RP	bC2	EA
Soybean	<b>84.43</b>	71.83	72.34	76.59
SPECT	<b>68.02</b>	61.11	56.28	56.55
ImgSeg	48.24	47.69	49.91	<b>51.31</b>
Heart	<b>68.26</b>	60.54	59.10	59.26
Wine	<b>79.19</b>	70.79	70.22	70.22
WDBC	<b>88.70</b>	85.41	85.38	85.41

Table 3:  $\rho_c$  for different datasets and methods (CF calculated when  $q = 1$ ).

We compare CF with three other cluster ensemble algorithms—bagged clustering (bC2, [11]), random projection (RP, [12]), and evidence accumulation (EA, [14]). We made slight modifications to the original implementations to standardize our comparison. These include adopting  $K$ -means clustering ( $K$ -medoids is used for bC2 in [11] but differs very little from  $K$ -means on the

datasets we have tried) to be the base clustering algorithm, changing the agglomerative algorithm used in RP to be based on single linkage in order to match the implementation in EA.

We now list the parameters used in our implementation. Define the number of initial clusters,  $n_b$ , to be that of clusters in running the base clustering algorithm; denote the number of final clusters by  $n_f$ . In CF, the scaling parameter  $\beta_1$  is set to be 10 (i.e., 0.1 times the ensemble size); the thresholding level  $\beta_2$  is 0.4 (we find very little difference in performance by setting  $\beta_2 \in [0.3, 0.5]$ ); the number of features,  $b$ , sampled each time in growing a clustering vector is 2; we set  $\tau_m = 3$  and  $n_b = n_f$ . In RP, the dimension of the target subspace for random projection is searched from 5 onwards and we set  $n_b = n_f$ . EA [14] suggests using  $\sqrt{n}$  ( $n$  being the sample size) for  $n_b$ . This sometimes leads to unsatisfactory results (which is the case for all except two of the datasets) and if that happens we replace it with  $n_f$ . In EA, the threshold value,  $t$ , for the single linkage algorithm is searched through  $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.75\}$  as suggested by [14]. In bC2, we set  $n_b = n_f$  according to [11].

Table 2 and Table 3 show the values of  $\rho_r$  and  $\rho_c$  achieved on the six UC Irvine datasets using RP, bC2, EA and CF. The ensemble size is 100 and results averaged over 100 runs. We take  $q = 1$  for CF in producing these two tables. We see that CF compares favorably to its competitors; it yields the largest  $\rho_r$  (or  $\rho_c$ ) for all but one of the six datasets and the performance gain is substantial in most cases (i.e., 4 out of 6).

We also explore the feature competition mechanism in the very initial round of CF (cf. Section 2.1). According to Theorem 1, in cases where there are many noise features or weak features, feature competition will decrease the chance of obtaining a weak clustering instance, hence a boost in the ensemble performance can be expected. In Table 5 and Table 4, we report results by varying the value of  $q$  in the feature competition step.

We define the *feature profile plot* to be the histogram of the strengths of each individual feature, where feature strength is defined as the  $\kappa$  value computed on the dataset using this feature alone. (For categorical variable when the number of categories on this variable is smaller than the number of clusters, the strength of this feature is sampled at random from the set of strengths of other features.) Figure 4 is the feature profile plot of the six UC Irvine datasets used in our experiment. A close inspection of results presented in Table 5 and Table 4 show that this plot can roughly guide us in choosing a  $q$  that leads to “good” performance for each individual dataset. We thus propose the following rule of thumb based on the feature profile plot: use large  $q$  when there are many weak or noise features; otherwise small  $q$  or no feature competition at all.

## 6 Conclusion

We have proposed a new method for ensemble-based clustering. Our experiments show that CF compares favorably to existing clustering ensemble methods, including bC2, evidence accumulation and RP. We have provided support-

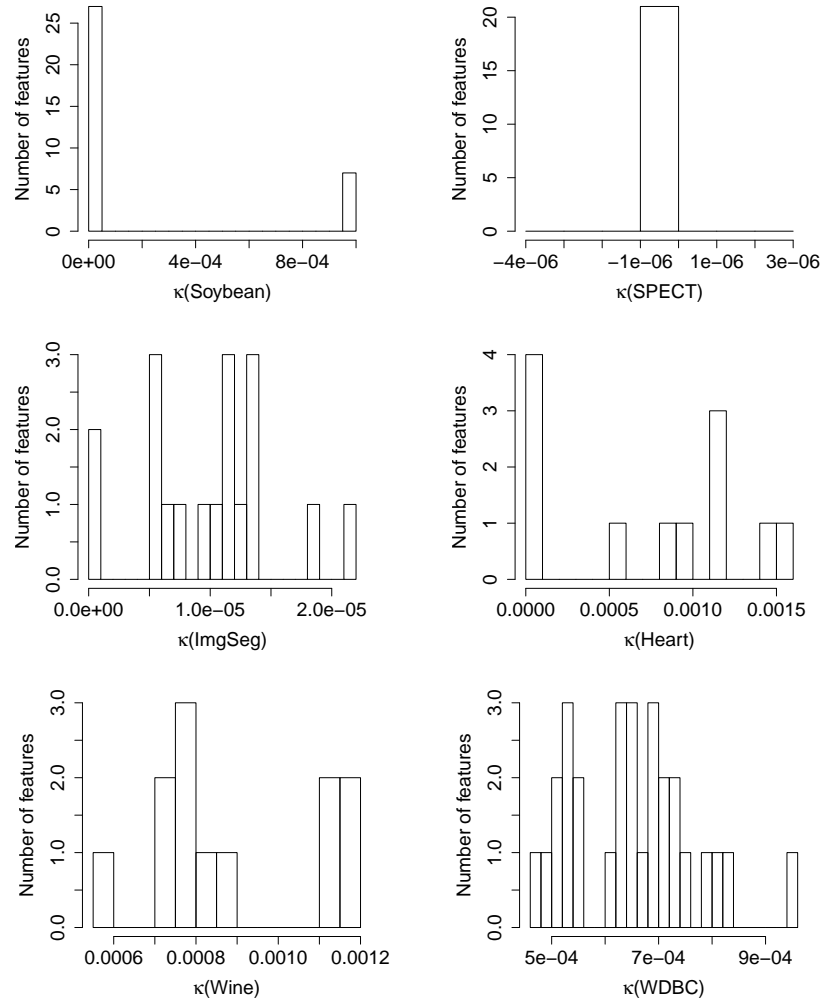


Figure 4: The feature profile plot for the 6 UC Irvine datasets.

$q$	Soybean	SPECT	ImgSeg	Heart	Wine	WDBC
1	92.36	56.78	79.71	56.90	<b>79.70</b>	<b>79.93</b>
2	92.32	57.39	77.62	60.08	74.02	<b>79.94</b>
3	94.42	57.24	77.51	62.51	72.16	79.54
5	93.89	<b>57.48</b>	81.17	63.56	71.87	79.41
10	93.14	56.54	82.69	<b>63.69</b>	71.87	78.90
15	94.54	55.62	<b>83.10</b>	<b>63.69</b>	71.87	78.64
20	<b>94.74</b>	52.98	82.37	<b>63.69</b>	71.87	78.50

Table 4: The  $\rho_r$  achieved by CF for  $q \in \{1, 2, 3, 5, 10, 15, 20\}$ . Results averaged over 100 runs. Note the first row is taken from Table 2.

$q$	Soybean	SPECT	ImgSeg	Heart	Wine	WDBC
1	84.43	68.02	48.24	68.26	<b>79.19</b>	<b>88.70</b>
2	84.91	<b>68.90</b>	43.41	72.20	72.45	<b>88.71</b>
3	89.85	68.70	41.12	74.93	70.52	88.45
5	89.13	68.67	47.92	76.13	70.22	88.37
10	88.40	66.99	49.77	<b>76.30</b>	70.22	88.03
15	90.96	65.15	49.65	<b>76.30</b>	70.22	87.87
20	<b>91.91</b>	60.87	<b>52.79</b>	<b>76.30</b>	70.22	87.75

Table 5: The  $\rho_c$  achieved by CF for  $q \in \{1, 2, 3, 5, 10, 15, 20\}$ . Results averaged over 100 runs. Note the first row is taken from Table 3.

ing theoretical analysis, showing that CF with  $\kappa$  is “noise-resistant” under a simplified model. We also obtain a closed-form formula for the mis-clustering rate of spectral clustering which yields new insights into the nature of spectral clustering, in particular it underscores the importance of the relative size of clusters to the performance of spectral clustering.

## 7 Appendix

In this appendix, Section 7.2 and Section 7.3 are devoted to the proof of Theorem 1 and Theorem 2, respectively. Section 7.1 deals with the equivalence, in the population, of the optimal clustering rule (as defined by equation (4) in Section 4.1 of the main text) and  $K$ -means clustering. This is to prepare for the proof of Theorem 1 and is of independent interest (e.g., it may help explain why  $K$ -means clustering may be competitive on certain datasets in practice).

### 7.1 Equivalence of $K$ -means clustering and the optimal clustering rule for mixture of spherical Gaussians

We first state and prove an elementary lemma for completeness.

**Lemma 1.** *For the Gaussian mixture model defined by (2) (Section 4.1) with  $\Sigma = \mathbf{I}$  and  $\pi = 1/2$ , in the population the decision rule induced by  $K$ -means clustering (in the sense of Pollard) is equivalent to the optimal rule  $h^*$  as defined in (4) (Section 4.1).*

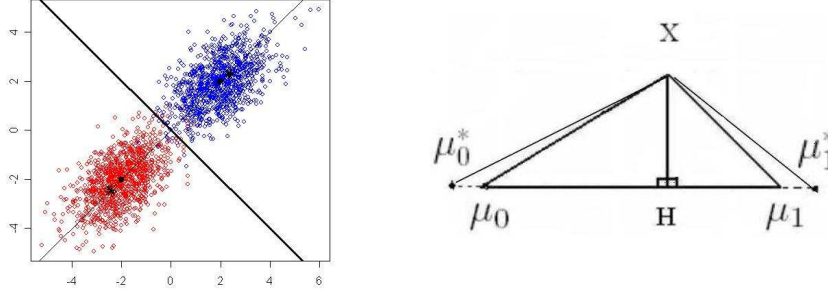


Figure 5: *The optimal rule  $h^*$  and the  $K$ -means rule. In the left panel, the decision boundary (the thick line) by  $h^*$  and that by  $K$ -means completely overlap for a 2-component Gaussian mixture with  $\Sigma = c\mathbf{I}$ . The stars in the figure indicate the population cluster centers by  $K$ -means. The right panel illustrates the optimal rule  $h^*$  and the decision rule by  $K$ -means where  $K$ -means compares  $\|X - \mu_0^*\|$  against  $\|X - \mu_1^*\|$  while  $h^*$  compares  $\|H - \mu_0\|$  against  $\|H - \mu_1\|$ .*

*Proof.* The geometry underlying the proof is shown in Figure 5. Let  $\mu_0, \Sigma_0$  and  $\mu_1, \Sigma_1$  be associated with the two mixture components in (2). By shift-invariance and rotation-invariance (rotation is equivalent to an orthogonal transformation which preserves clustering membership for distance-based clustering), we can reduce to the  $\mathbb{R}^1$  case such that  $\mu_0 = (\mu_0[1], 0, \dots, 0) = -\mu_1$  with  $\Sigma_0 = \Sigma_1 = \mathbf{I}$ . The rest of the argument follows from geometry and the definition of  $K$ -means clustering, which assigns  $X \in \mathbb{R}^d$  to class 1 if  $\|X - \mu_1^*\| < \|X - \mu_0^*\|$ , and the optimal rule  $h^*$  which determines  $X \in \mathbb{R}^d$  to be in class 1 if

$$(\mu_1 - \mu_0)^T \left( X - \frac{\mu_0 + \mu_1}{2} \right) > 0,$$

or equivalently,

$$\|X - \mu_1\| < \|X - \mu_0\|.$$

□

## 7.2 Proof of Theorem 1

*Proof of Theorem 1.* Let  $\mathcal{C}_1$  and  $\mathcal{C}_0$  denote the two clusters obtained by  $K$ -means clustering. Let  $G$  be the distribution function of the underlying data.



$SS_W$  and  $SS_B$  can be calculated as follows.

$$\begin{aligned} SS_W &= \frac{1}{2} \int_{x \neq y \in \mathcal{C}_1} \|x - y\|^2 dG(x) dG(y) + \frac{1}{2} \int_{x \neq y \in \mathcal{C}_0} \|x - y\|^2 dG(x) dG(y) \triangleq (\sigma_d^*)^2, \\ SS_B &= \int_{x \in \mathcal{C}_1} \int_{y \in \mathcal{C}_0} \|x - y\|^2 dG(y) dG(x) = (\sigma_d^*)^2 + \frac{1}{4} \|\mu_0^* - \mu_1^*\|^2. \end{aligned}$$

If we assume  $\Sigma = \mathbf{I}_{p \times p}$  is always true during the growth of the clustering vector (this holds if duplicated features are excluded), then

$$\frac{1}{\kappa} = \frac{SS_B}{SS_W} = 1 + \frac{\|\mu_0^* - \mu_1^*\|^2}{4(\sigma_d^*)^2}. \quad (11)$$

Without loss of generality, let  $\mathcal{I} = \{1, 2, \dots, d-1\}$  and let the noise feature be the  $d^{th}$  feature. By the equivalence, in the population, of  $K$ -means clustering and the optimal clustering rule  $h^*$  (Lemma 1 in Section 7.1) for a mixture of two spherical Gaussians,  $K$ -means clustering assigns  $x \in \mathbb{R}^d$  to  $\mathcal{C}_1$  if

$$\|x - \mu_1\| < \|x - \mu_0\|,$$

which is equivalent to

$$\sum_{i=1}^{d-1} (x[i] - \mu_1[i])^2 < \sum_{i=1}^{d-1} (x[i] - \mu_0[i])^2. \quad (12)$$

This is true since  $\mu_0[d] = \mu_1[d] = 0$  by the assumption that the  $d^{th}$  feature is a noise feature. (12) implies that the last coordinate of the population cluster centers for  $\mathcal{C}_1$  and  $\mathcal{C}_2$  are the same, that is,  $\mu_1^*[d] = \mu_0^*[d]$ . This is because, by definition,  $\mu_i^*[j] = \int_{x \in \mathcal{C}_i} x[j] dG(x)$  for  $i = 0, 1$  and  $j = 1, \dots, d$ . Therefore adding a noise feature does not affect  $\|\mu_0^* - \mu_1^*\|^2$ . However, the addition of a noise feature would increase the value of  $(\sigma_d^*)^2$ , it follows that  $\kappa$  will be increased by adding a noise feature.  $\square$

### 7.3 Proof of Theorem 2

*Proof of Theorem 2.* To simplify the presentation, some lemmas used here (Lemma 2 and Lemma 3) are stated after this proof.

It can be shown that

$$D^{-1/2} \overline{P} D^{-1/2} = \sum_{i=1}^2 \lambda_i \mathbf{x}_i \mathbf{x}_i^T,$$

where  $\lambda_i$  are the eigenvalues and  $\mathbf{x}_i$  eigenvectors,  $i = 1, 2$ , such that for  $\nu = o(1)$ ,

$$\begin{aligned} \lambda_1 &= 1 + O_p(\nu^2 + n^{-1}), \\ \lambda_2 &= 1 - \gamma^{-1}(1 + \gamma^2)\nu + O_p(\nu^2 + n^{-1}) \end{aligned}$$

and

$$(n_1\gamma^{-1} + n_2)^{1/2}\mathbf{x}_1[i] = \begin{cases} \gamma^{-1/2} + O_p(\nu + n^{-1/2}), & \text{if } i \leq n_1 \\ 1 + O_p(\nu + n^{-1/2}), & \text{otherwise.} \end{cases}$$

$$(n_1\gamma^3 + n_2)^{1/2}\mathbf{x}_2[i] = \begin{cases} -\gamma^{3/2} + O_p(\nu + n^{-1/2}), & \text{if } i \leq n_1 \\ 1 + O_p(\nu + n^{-1/2}), & \text{otherwise.} \end{cases}$$

By Lemma 3, we have  $\|\varepsilon\|_2 = O_p(\sqrt{n})$  and thus the  $i^{th}$  eigenvalues of  $D^{-1/2}PD^{-1/2}$  for  $i \geq 3$  are of order  $O_p(n^{-1/2})$ . Note that, in the above, all residual terms are uniformly bounded w.r.t.  $n$  and  $\nu$ .

Let

$$\psi = \frac{1}{2\pi i} \int_{\Gamma} (tI - D^{-1/2}PD^{-1/2})^{-1} dt,$$

where  $\Gamma$  is a Jordan curve enclosing only the first two eigenvalues. Then, by (8) and (9) in the main text (see Section 4.2),  $\psi\mathbf{x}_2$  is the second eigenvector of  $D^{-1/2}PD^{-1/2}$  and the mis-clustering rate is given by

$$\mathcal{M} = \frac{1}{n} \left[ \sum_{i \leq n_1} I((\psi\mathbf{x}_2)[i] < 0) + \sum_{i > n_1} I((\psi\mathbf{x}_2)[i] > 0) \right].$$

Thus

$$\mathbb{E}\mathcal{M} = \frac{1}{1+\gamma} [\mathbb{P}((\psi\mathbf{x}_2)[i] > 0) + \gamma\mathbb{P}((\psi\mathbf{x}_2)[i] < 0)].$$

By Lemma 2 and letting  $\tilde{\varepsilon} = D^{-1/2}\varepsilon D^{-1/2}$ , we have

$$\begin{aligned} \psi\mathbf{x}_2 &= \frac{1}{2\pi i} \int_{\Gamma} (tI - D^{-1/2}\overline{P}D^{-1/2} - \tilde{\varepsilon})^{-1} \mathbf{x}_2 dt \\ &= \frac{1}{2\pi i} \int_{\Gamma} (I - (tI - D^{-1/2}\overline{P}D^{-1/2})^{-1}\tilde{\varepsilon})^{-1} (tI - D^{-1/2}\overline{P}D^{-1/2})^{-1} \mathbf{x}_2 dt \\ &= \frac{1}{2\pi i} \int_{\Gamma} (I - (tI - D^{-1/2}\overline{P}D^{-1/2})^{-1}\tilde{\varepsilon})^{-1} \mathbf{x}_2 (t - \lambda_2)^{-1} dt \\ &= \phi\mathbf{x}_2 + O_p(n^{-2}), \end{aligned}$$

where

$$\phi\mathbf{x}_2 = \frac{1}{2\pi i} \int_{\Gamma} [I + (tI - D^{-1/2}\overline{P}D^{-1/2})^{-1}\tilde{\varepsilon}] \mathbf{x}_2 (t - \lambda_2)^{-1} dt.$$

It can be shown that, by the Cauchy Integral Theorem [33] and Lemma 2,

$$\begin{aligned} \phi\mathbf{x}_2 &= \mathbf{x}_2 + \frac{1}{2\pi i} \int_{\Gamma} (tI - D^{-1/2}\overline{P}D^{-1/2})^{-1} \tilde{\varepsilon}\mathbf{x}_2 (t - \lambda_2)^{-1} dt. \\ &= \mathbf{x}_2 - \lambda_2^{-1}\tilde{\varepsilon}\mathbf{x}_2 + O_p(n^{-2}). \end{aligned}$$

Let  $\tilde{\varepsilon}_i$  be the  $i^{th}$  column of  $\tilde{\varepsilon}$ . By Slutsky's Theorem, one can verify that

$$\tilde{\varepsilon}_1^T \mathbf{x}_2 = \sigma(n_1 n_2)^{-1/2} \mathcal{N}\left(0, \frac{1+\gamma^3}{1+\gamma^2}\right) + O_p(n^{-2})$$

and

$$\tilde{\varepsilon}_n^T \mathbf{x}_2 = n_2^{-1} \sigma \mathcal{N}\left(0, \frac{1+\gamma^3}{1+\gamma^2}\right) + O_p(n^{-2}).$$

Thus

$$\begin{aligned} \mathbb{P}((\psi \mathbf{x}_2)[1] < 0) &= \mathbb{P}\left(\mathcal{N}(0, 1) > (n_1 n_2)^{1/2} \sigma^{-1} \sqrt{\frac{1+\gamma^2}{1+\gamma^3}} \frac{\gamma^{3/2}}{\sqrt{n_1 \gamma^3 + n_2}}\right) (1 + o(1)) \\ &= \mathbb{P}\left(\mathcal{N}(0, 1) > n^{1/2} \sigma^{-1} \sqrt{\frac{\gamma^2}{(1+\gamma)(1+\gamma^3)}}\right) (1 + o(1)) \end{aligned}$$

and

$$\begin{aligned} \mathbb{P}((\psi \mathbf{x}_2)[1] > 0) &= \mathbb{P}\left(\mathcal{N}(0, 1) > n_2 \sigma^{-1} \sqrt{\frac{1+\gamma^2}{1+\gamma^3}} \frac{1}{\sqrt{n_1 \gamma^3 + n_2}}\right) (1 + o(1)) \\ &= \mathbb{P}\left(\mathcal{N}(0, 1) > n^{1/2} \sigma^{-1} \sqrt{\frac{\gamma}{(1+\gamma)(1+\gamma^3)}}\right) (1 + o(1)). \end{aligned}$$

Hence

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log(\mathbb{E} \mathcal{M}) = -\frac{\gamma^2}{2\sigma^2(1+\gamma)(1+\gamma^3)},$$

and the conclusion follows.  $\square$

**Lemma 2.** Let  $\overline{P}, \mathbf{x}_2, \lambda_2, \psi, \phi$  be defined as above. Then

$$\left(tI - D^{-1/2} \overline{P} D^{-1/2}\right)^{-1} \mathbf{x}_2 = (t - \lambda_2)^{-1} \mathbf{x}_2$$

and

$$\|\psi \mathbf{x}_2 - \phi \mathbf{x}_2\|_\infty = O_p(n^{-2}).$$

The first part follows from a direct calculation and the proof of the second relies on the semi-circle law. The technical details are omitted.

**Lemma 3.** Let  $\varepsilon = \{\varepsilon_{ij}\}_{i,j=1}^n$  be a symmetric random matrix with  $\varepsilon_{ij} \sim \mathcal{N}(0, 1)$ , independent for  $1 \leq i \leq j \leq n$ . Then

$$\|\varepsilon\|_2 = O_p(\sqrt{n}).$$

The proof is based on the moment method (see [16]) and the details are omitted.

## References

- [1] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *Eighteenth Annual Conference on Computational Learning Theory (COLT)*, 2005.
- [2] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.
- [3] A. Asuncion and D. Newman. UC Irvine Machine Learning Repository. <http://www.ics.uci.edu/mllearn/MLRepository.html>, 2007.
- [4] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *ACM Symposium on the Theory of Computing*, 2001.
- [5] J. Azimi and X. Fern. Adaptive cluster ensemble selection. In *International Joint Conference on Artificial Intelligence*, 2009.
- [6] P. Bickel and A. Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.
- [7] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.
- [8] R. Caruana, A. Niculescu, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *The International Conference on Machine Learning*, 2004.
- [9] S. Dasgupta. Experiments with random projection. In *Uncertainty in Artificial Intelligence: Proceedings of the 16th Conference (UAI)*, 2000.
- [10] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [11] S. Dudoit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099, 2003.
- [12] X. Fern and C. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003.
- [13] X. Fern and W. Lin. Cluster ensemble selection. *Statistical Analysis and Data Mining*, 1:128–141, 2008.
- [14] A. Fred and A. Jain. Data clustering using evidence accumulation. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 276–280, 2002.
- [15] A. Fred and A. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

- [16] Z. Furedi and J. Komlos. The eigenvalues of random symmetric matrices. *Combinatorica*, 1:233–241, 1981.
- [17] A. Gionis, H. Mannila, and P. Tsaparas. Cluster aggregation. In *International Conference on Data Engineering*, 2005.
- [18] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2009.
- [19] C. Hegde, M. Wakin, and R. Baraniuk. Random projections for manifold learning. In *Neural Information Processing Systems (NIPS)*, volume 20, 2007.
- [20] P. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: Some first steps. *Social Networks*, 5:109–137, 1983.
- [21] L. Huang, D. Yan, M. I. Jordan, and N. Taft. Spectral clustering with perturbed data. In *Neural Information Processing Systems (NIPS)*, volume 21, 2009.
- [22] P. Huber. Projection pursuit. *Annals of Statistics*, 13:435–475, 1985.
- [23] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [24] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.
- [25] T. Kato. *Perturbation Theory for Linear Operators*. Springer, Berlin, 1966.
- [26] M. Meila and J. Shi. Learning segmentation by random walks. In *Neural Information Processing Systems (NIPS)*, 2001.
- [27] M. Meila, S. Shortreed, and L. Xu. Regularized spectral learning. Technical report, Department of Statistics, University of Washington, 2005.
- [28] B. Minaei, A. Topchy, and W. Punch. Ensembles of partitions via data resampling. In *Proceedings of International Conference on Information Technology*, 2004.
- [29] B. Nadler, S. Lafon, and R. Coifman. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In *Neural Information Processing Systems (NIPS)*, volume 18, 2005.
- [30] A. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In *Neural Information Processing Systems (NIPS)*, volume 14, 2002.
- [31] K. Nowicki and T. Snijders. Estimation and prediction for stochastic block-structures. *Journal of the American Statistical Association*, 96:1077–1087, 2001.

- [32] D. Pollard. Strong consistency of K-means clustering. *Annals of Statistics*, 9(1):135–140, 1981.
- [33] W. Rudin. *Real and Complex Analysis (Third Edition)*. McGraw-Hill, 1986.
- [34] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [35] T. Shi and S. Horvath. Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 15(1):118–138, 2006.
- [36] A. Strehl and J. Ghosh. Cluster ensembles: A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:582–617, 2002.
- [37] A. Topchy, A. Jain, and W. Punch. Combining multiple weak clusterings. In *Proceedings of the IEEE International Conference on Data Mining*, 2003.
- [38] S. Vempala and G. Wang. A spectral algorithm for learning mixture models. *J. Comput. Syst. Sci.*, 68:841–860, 2004.
- [39] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.
- [40] U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *Annals of Statistics*, 36(2):555–586, 2008.
- [41] D. Yan, L. Huang, and M. I. Jordan. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 907–916, 2009.
- [42] Z. Zhang and M. I. Jordan. Multiway spectral clustering: A margin-based perspective. *Statistical Science*, 23:383–403, 2008.